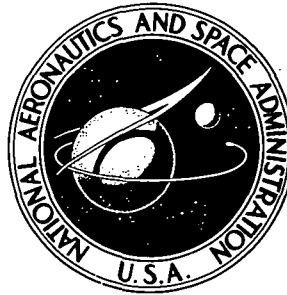


NASA TECHNICAL NOTE



NASA TN D-7352

NASA TN D-7352

(NASA-TN-D-7352) BEAMR: AN INTERACTIVE  
GRAPHIC COMPUTER PROGRAM FOR DESIGN OF  
CHARGED PARTICLE BEAM TRANSPORT SYSTEMS  
(NASA) 45 p HC \$3.00 CSCL 09B

N73-27143

Unclass

H1/08 08842

BEAMR - AN INTERACTIVE  
GRAPHIC COMPUTER PROGRAM  
FOR DESIGN OF CHARGED-PARTICLE  
BEAM TRANSPORT SYSTEMS

*by Regis F. Leonard and Charles C. Giamati*

*Lewis Research Center*

*Cleveland, Ohio 44135*

1. Report No. <b>NASA TN D-7352</b>		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle <b>BEAMR - AN INTERACTIVE GRAPHIC COMPUTER PROGRAM FOR DESIGN OF CHARGED- PARTICLE BEAM TRANSPORT SYSTEMS</b>				5. Report Date <b>July 1973</b>	
				6. Performing Organization Code	
7. Author(s) <b>Regis F. Leonard and Charles C. Giamati</b>				8. Performing Organization Report No. <b>E-7384</b>	
9. Performing Organization Name and Address <b>Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135</b>				10. Work Unit No. <b>503-10</b>	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address <b>National Aeronautics and Space Administration Washington, D. C. 20546</b>				13. Type of Report and Period Covered <b>Technical Note</b>	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract  A computer program for a PDP-15 is presented which calculates, to first order, the characteristics of a charged-particle beam as it is transported through a sequence of focusing and bending magnets. The maximum dimensions of the beam envelope normal to the transport system axis are continuously plotted on an oscilloscope as a function of distance along the axis. Provision is made to iterate the calculation by changing the types of magnets, their positions, and their field strengths. The program is especially useful for transport system design studies because of the ease and rapidity of altering parameters from panel switches. A typical calculation for a system with eight elements is completed in less than 10 seconds. An IBM 7094 version containing more-detailed printed output but no oscilloscope display is also presented.					
17. Key Words (Suggested by Author(s)) <b>Computer program; Interactive graphic; Charged-particle beam transport; Optics; PDP-15; Bending magnet</b>				18. Distribution Statement <b>Unclassified - unlimited</b>	
19. Security Classif. (of this report) <b>Unclassified</b>		20. Security Classif. (of this page) <b>Unclassified</b>		21. No. of Pages <b>44</b>	
				22. Price* <b>\$3.00</b>	

# BEAMR - AN INTERACTIVE GRAPHIC COMPUTER PROGRAM FOR DESIGN OF CHARGED-PARTICLE BEAM TRANSPORT SYSTEMS

by Regis F. Leonard and Charles C. Giamati

Lewis Research Center

## SUMMARY

A computer program is presented which calculates the characteristics of a charged-particle beam as it is transported through a sequence of focusing and bending magnets. The maximum dimensions of the beam envelope normal to the transport system axis are continuously plotted on an oscilloscope as a function of distance along the axis. Provision is made to iterate the calculation by changing the types of magnets, their positions, and their field strengths. The program is especially useful for transport system design studies because of the ease and rapidity of altering parameters from panel switches. A typical calculation for a system with eight elements is completed in less than 10 seconds. A version containing more-detailed printed output but no oscilloscope display is also presented.

## INTRODUCTION

Designing a beam-handling system for the experimental facility associated with the modified NASA Lewis 60-inch cyclotron required that we calculate the effect of each proposed beam-handling element upon the beam properties. In fact, calculating the properties of the beam at each point in the system from the source (the cyclotron) to the target was necessary in order to determine not only the characteristics and physical locations of the beam-handling equipment, but also the field-settings which would provide the highest quality beam for a particular experiment after the hardware had been placed.

Four characteristics of a charged-particle beam are most significant to a prospective user. The first is the spatial extent of the beam ( $x, y$ ), both as it traverses the beam-handling system and as it arrives at the target. Closely related to the spatial extent, and for many experiments of equal importance, is the divergence of the beam ( $x', y'$ ) when it reaches the target. In addition, it is often necessary to

know the spread in momentum  $\Delta p$  of the particles contained in the beam. For experiments which involve timing, it is also necessary to have some knowledge of the spread in time (anisochronicity  $\Delta\lambda$ ) of the incident beam. Thus, a complete description of the beam, at least for purposes of any experiment presently anticipated, requires six dimensions.

## MATHEMATICAL METHOD

### Single Charged Particle

The basic assumption underlying all calculations performed by the present program is that the transformation of the coordinates of a particle at any point in the system to a later point is linear (i.e., the calculations are carried out to first order). The transformation may be written

$$\bar{X}_1 = \mathcal{T}\bar{X}_0$$

where  $\bar{X}_1$  is a six-dimensional vector whose components are the coordinates of a particle at time (or position) 1,  $\bar{X}_0$  is a vector comprised of the set of these same coordinates at time (or position) 0, and  $\mathcal{T}$  is the transformation matrix. The scalar coordinates selected for description of a single particle traversing the beam-handling system are

- x      distance of particle from system centerline in the horizontal plane
- x'     angle between system centerline and projection of particle momentum onto  
         horizontal plane;  $x' = dx/dz$ , where  $z$  is coordinate along system axis
- $\Delta p$    deviation of particle momentum from that of particle which would traverse  
         system centerline
- $\Delta\lambda$    deviation in distance traveled from that of particle traversing system center-  
         line
- y      distance of particle from system centerline in vertical plane
- y'     angle between system centerline and projection of particle momentum onto  
         vertical plane;  $y'$  is analogous to  $x'$

The coordinates  $x$ ,  $x'$ ,  $y$ ,  $y'$  are shown graphically in figure 1.

The problem then becomes one of determining how each type of beam-handling element encountered will transform the particle coordinates (i.e., what is the nature

of the transformation matrix  $\mathcal{T}$  for each piece of hardware). These matrices are well known (refs. 1 to 3) and are easily constructed. For the convenience of the reader, they are summarized here as follows:

(1) For a field-free region (drift space) of length  $L$ , the  $\mathcal{T}$  matrix is given by

$$\mathcal{T} = \begin{pmatrix} 1 & L & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & L \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Hence,  $\bar{X}_1 = \mathcal{T}\bar{X}_0$  is equivalent to the scalar equations

$$x_1 = x_0 + Lx'_0$$

$$x'_1 = x'_0$$

$$\Delta p_1 = \Delta p_0$$

$$\Delta\lambda_1 = \Delta\lambda_0$$

$$y_1 = y_0 + Ly'_0$$

$$y'_1 = y'_0$$

(2) For a uniform-field magnet, the  $\mathcal{T}$  matrix is given by

$$\begin{pmatrix}
\frac{\cos(\varphi - \alpha)}{\cos \alpha} & R \sin \varphi & R(1 - \cos \varphi) & 0 & 0 & 0 \\
-\frac{\sin(\varphi - \alpha - \beta)}{R \cos \alpha \cos \beta} & \frac{\cos(\varphi - \beta)}{\cos \beta} & \sin \varphi + \tan \beta(1 - \cos \varphi) & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
\sin \varphi + \tan \alpha(1 - \cos \varphi) & R(1 - \cos \varphi) & R(\varphi - \sin \varphi) & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 - \varphi \tan \alpha & R\varphi \\
0 & 0 & 0 & 0 & -\tan \alpha - \tan \beta + \varphi \tan \alpha \tan \beta & 1 - \varphi \tan \beta
\end{pmatrix}$$

where  $R$ ,  $\varphi$ ,  $\alpha$ , and  $\beta$  are magnet dimensions, defined by figure 2.

(3) For a quadrupole lens - horizontally defocusing, vertically focusing (ref. 2), the matrix is given by

$$\begin{pmatrix}
\cosh(kL) & \frac{1}{k} \sinh(kL) & 0 & 0 & 0 & 0 \\
k \sinh(kL) & \cosh(kL) & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos(kL) & \frac{1}{k} \sin(kL) \\
0 & 0 & 0 & 0 & -k \sin(kL) & \cos(kL)
\end{pmatrix}$$

where  $L$  is the effective length of the lens, and  $k$  is the focusing strength, related to the magnetic field gradient  $G$  by

$$k^2 = eG/p$$

where  $p/e$  is the magnetic rigidity. If  $p/e$  is expressed in kilogauss-inches, and  $G$  in kilogauss/inches, then  $k$  has the dimensions  $(\text{in.})^{-1}$ .

(4) For a quadrupole lens - horizontally focusing, vertically defocusing (ref. 2), the matrix is given by

$$\begin{pmatrix} \cos(kL) & \frac{1}{k} \sin(kL) & 0 & 0 & 0 & 0 \\ -k \sin(kL) & \cos(kL) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cosh(kL) & \frac{1}{k} \sinh(kL) \\ 0 & 0 & 0 & 0 & k \sinh(kL) & \cosh(kL) \end{pmatrix}$$

(5) For a rotation about the beam axis (ref. 3), the matrix is given by

$$\begin{pmatrix} \cos \theta & 0 & 0 & 0 & \sin \theta & 0 \\ 0 & \cos \theta & 0 & 0 & 0 & \sin \theta \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & 0 & 0 & \cos \theta & 0 \\ 0 & -\sin \theta & 0 & 0 & 0 & \cos \theta \end{pmatrix}$$

where  $\theta$  is the angle of rotation - positive for a clockwise (looking downstream) rotation, negative for counterclockwise (looking downstream) rotation. The rotations are mathematical operations in the program which rotate the beam in order to use a single bending magnet matrix for all physical orientations of the magnets.

### Beam of Particles

The formulation just given is sufficient for the calculation of the trajectory of a single charged particle through a system of magnets. However, the physical entity to be treated is a collection of particles with a spread in initial coordinates. In one dimension, for example, the beam would occupy a certain region of the  $x, x'$  space.

This is most conveniently represented by a phase-space ellipse, as shown in figure 3. At a waist, or spatial minimum, the phase-space ellipse becomes upright. As the beam traverses a field-free region, for example, the ellipse becomes skewed, as shown by the third ellipse in figure 3.

The formulation of the transformation of a beam envelope or ellipse is again most easily accomplished by means of a matrix formalism (ref. 3). If the beam envelope at a time (or position) 0 is given by

$$axx + bxx' + cx'x' = 1 \quad (1)$$

then the envelope at time (or position) 1 will be given by a similar ellipse but with coefficients  $a'$ ,  $b'$ , and  $c'$ . In matrix form, equation (1) may be written

$$\begin{pmatrix} x & x' \end{pmatrix} \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix} = 1$$

or

$$\bar{X}^T \mathcal{S}^{-1} \bar{X} = 1$$

It is easily shown that a matrix  $\mathcal{T}$  which transforms the coordinates  $x, x'$  to new values will transform the matrix  $\mathcal{S}$  as

$$\mathcal{S}(2) = \mathcal{T} \mathcal{S}(1) \mathcal{T}^T$$

Furthermore, the physical dimensions of the beam are related to the diagonal elements of the  $\mathcal{S}$  matrix:

$$x_{\max} = (S_{11})^{1/2}$$

$$x'_{\max} = (S_{22})^{1/2}$$

The generalization of the method for a six-dimensional problem is straightforward.



The final transport to the target is accomplished by making the target element equivalent to a rotation of zero degrees preceded by the actual drift length.

The program flow is diagrammed in figure 4. Initially, the input parameters are typed in for the following quantities (see the section Input Values for details):

- (1) The initial beam vector, XMAX
- (2) The number of elements, JMAX
- (3) The parameters for each element

The parameters XMAX and JMAX are read in by the main program. Each set of parameters for an element is read in by the subroutine REED; the transport matrix for that element is written, if desired, by the subroutine RITE. The matrices are calculated by the subroutines MAG, QUAD, and ROT.

For the transport calculation, the index J is incremented by 1 until JMAX is reached. For a given J, the subroutine DRIFT calculates the values of  $x_{\max}$ ,  $y_{\max}$  in the drift space. The total transport matrix  $\mathcal{T}$ , representing the entire system up to the given point, is updated after the return from DRIFT. The transport matrix is again updated by MATMLT, which multiplies  $\mathcal{T}$  by the proper element matrix; and these values of the  $\mathcal{T}$  matrix (at the end of an element) are saved.

At the end of the calculation, the DSPLA subroutine displays the results and pauses (Pause 2) until a further command is entered from the panel (fig. 5) or until the light-pen switch is pressed. The oscilloscope display (fig. 6) includes points representing  $x_{\max}$  and  $y_{\max}$  as a function of distance z from the entrance of the transport system (see the section Display). If the light-pen is held over a selected point and the pen switch depressed (initiating a call to the subroutine LANDT), the  $\mathcal{T}$  matrix and the beam vector XMAX at that point will be printed on the Teletypewriter. Following Pause 2, the calculation will be repeated or halted according to the setting of the repeat toggle switch. In either case, the values of  $x_{\max}$  and  $y_{\max}$  will be printed on the Teletypewriter if the print toggle switch is set "up." The values printed are for a range of J as selected on the rotary switches. The subroutine OUTPT accomplishes this printing. Pressing the continue button during Pause 2 resumes the processing according to the options set on the panel.

If the calculation is to be repeated, more input is required. If only the magnetic field strengths for two quadrupoles are to be changed, this can be done from the panel before the continue button is pressed. For changing other parameters, more input through the Teletypewriter is required. A (possibly new) value of JMAX is

input, along with the number of the element whose parameters, and possibly whose type, are to be changed. After the input for an element, the value of the variable AGAIN is set to TRUE if another element is to be changed, and is set to FALSE otherwise. In the program, the subroutine REPET controls the input and repeats the calculation.

When the calculation is halted, either the present case can be dumped or a new calculation can be initiated. Pushbutton D selects the dump. Pushbutton 7 initiates the new calculation, which then requires a new input of the beam parameters.

If a program has been dumped, the calculations can be resumed at a later time. The subroutine RFRESH provides an option for printing out the parameter values currently stored in the program when it is being reloaded from a prior dump. The value of the variable PRINT is entered as TRUE on the Teletypewriter if the printout is wanted, FALSE otherwise.

### Display

The oscilloscope display (fig. 6) consists of

(1) Values of  $x_{\max}$  and  $y_{\max}$  as function of distance  $z$ , where  $x_{\max}$  is plotted as a positive value and  $y_{\max}$  as a negative value from the centerline (The values are displayed at 10 equally spaced points in each drift space.)

(2) Two horizontal lines, corresponding to the physical beam tube wall, a physical distance of 3.81 centimeters (1.5 in.) from the beam axis

(3) Labels for each element (M,Q,R,T) along with an element index number

(4) A magnified display of the target at the extreme right of the screen (Two fiducial lines each represent a distance of 1 mm from the beam centerline. The  $x_{\max}$  and  $y_{\max}$  of the beam at the target are displayed at the magnification corresponding to the fiducial lines.)

### Panel

The panel (fig. 5) consists of

(1) Lamps - Lamp 1 being on signals Pause 1 at the beginning of the program. The continue button is used to proceed. Lamp 2 being on signals Pause 2 at the completion of a new calculation and display. The continue button is used to proceed. Lamp 7 being on signals Pause 7 for the selection of the "Dump or go to 1" option. Either the dump or go-to-1 button is used to proceed.

**Page Intentionally Left Blank**

# APPENDIX A

## FORTRAN LISTING FOR PDP-15 VERSION

```

C . . . .MAIN PROGRAM FOR BEAM, BEAM TRANSPORT
C
      LOGICAL BUTTNS,HALT,TCGGLS,BITEST
      COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL
      COMMON/ELEMT/RO(15),PHI(15),ALP(15),BET(15),EL(15),FG(15),
      1ALPH(15),IQMR(15)
      COMMON/TAX/T(6,6),XMAX(6)
      COMMON/SAV/TSV(6,6,15),ZNOT(16)
      COMMON/DSPXY/XDSP(150),YDSP(150,2),JMAX
C . . . . .READY, INITIALIZE ARRAYS
101  CALL LAMPON(1)
      IBUT=0
110  IF(BUTTNS(IBUT)) GO TO 110
      IF(IBUT.LT.128) GO TO 110
      CALL ZERO(A)
      CALL ZERO(XDSP)
      CALL ZERO(YDSP)
      CALL LAMPCF(1)
C . . . . .INPUT BEAM,ELEMENT PARAMETERS
      CALL INLIST(XMAX,5H XMAX)
      CALL INPUT1(JMAX,4HJMAX,1)
      DO 115 J=1,JMAX
      CALL REED(J)
      CALL TOGGLS(ITOG)
      IF(.NOT.BITEST(ITOG,10)) GO TO 115
      CALL RITE(J)
115  CONTINUE
C . . . . .INITIAL VALUES FOR FIRST CALCULATION
      J=1
200  DO 205 M1=1,6
      DO 205 M2=1,6
      T(M1,M2)=0.
205  IF(M1.EQ.M2) T(M1,M2)=1.
C . . . . .BEGIN LOOP FOR TRANSPORT
251  DEL= ZL(J)/10.
      IZ=J*10 - 10
      ZTOT = ZNOT(J)
      IF(J.EQ.1) ZTOT = 0.
252  CALL DRIFT(DEL,ZTOT,IZ)

```

```

DO 255 L=1,6
T(1,L) = T(1,L) + T(2,L)*ZL(J)
255 T(5,L) = T(5,L) + T(6,L)*ZL(J)
ZTOT = ZTCT + ZL(J)
CALL MATMLT(J)
ZTOT = ZTCT + DIST(J)
JP1 = J+1
ZNCT(JP1) = ZTOT
DO 275 M1=1,6
DO 275 M2=1,6
275 TSV(M1,M2,J) = T(M1,M2)
J = JP1
299 IF(J.LE.JMAX) GO TO 251
C . . . . . END TRANSPORT LOOP
301 CALL DSPLA
CALL TOGGLS(ITOG)
IF(.NOT.BITEST(ITOG,14)) GO TO 402
400 CALL OUTPT
402 CALL TOGGLS(ITOG)
HALT=.TRUE.
IF(BITEST(ITOG,12)) HALT=.FALSE.
IF(HALT) GO TO 599
500 CALL REPET(JMIN)
J=JMIN
CALL TOGGLS(ITOG)
IF(.NOT.BITEST(ITOG,15)) GO TO 504
CALL CLKOFF
504 IF(J.EQ.1) GO TO 200
JM1 = J-1
DO 505 M1=1,6
DO 505 M2=1,6
505 T(M1,M2) = TSV(M1,M2,JM1)
GO TO 251
599 IE=0
CALL LAMPON(7)
600 IF(BUTTNS(IE)) GO TO 600
IF(IB.NE.16.AND.IB.NE.32) GO TO 600
CALL LAMPCF(7)
IF(IB.EQ.16) GO TO 601
GO TO 101
601 CALL VPSTOP
CALL QDUMF
CALL VPSTRT
CALL RFRESH
GO TO 301
END

```

PRECEDING PAGE BLANK NOT FILMED

COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL  
COMMON/ELEMT/RO(15),PHI(15),ALP(15),BET(15),EL(15),PG(15),  
IALPH(15),IQMR(15)

C

JW = J  
IEL=IQMR(JW)  
WRITE(6,10) JW,IEL,ZL(J),DIST(J)  
10 FORMAT(1H0,17H ELEMENT NUMBER =,I1,7H TYPE= ,I1,12H  
PRE-DRIFT= ,1F10.5,11H LENGTH =,F10.5/)  
DO 115 K=1,6  
WRITE(6,20) (A(K,N,J),N=1,6)  
115 CONTINUE  
RETURN  
20 FORMAT(1H ,F11.6,5F12.6)  
END

SUBROUTINE MAG(J,RO,PHI,ALP,BET)

C

C . . . . COMPUTE MAGNET MATRIX FROM INPUT PARAMETERS

COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL

C . . . . .SET ELEMENTS USED IN QUAD OR ROT TO ZERO

A(1,5,J) = 0.

A(2,6,J) = 0.

A(5,1,J) = 0.

A(6,2,J) = 0.

C . . . . COMPUTE SINE AND CCSINE VALUES

RPI=3.141593/180.

PR=PHI\*RPI

AR=ALP\*RPI

BR=BET\*RPI

CP=COS(PR)

CA=COS(AR)

CB=COS(BR)

SP=SIN(PR)

SA=SIN(AR)

SB=SIN(BR)

TA=SA/CA

TB=SB/CB

A(1,1,J) = CP+SP\*TA

A(1,2,J) = RO\*SP

A(1,3,J) = RO\*(1.-CP)

A(2,1,J) = -SIN(PR-AR-BR)/(RO\*CA\*CB)

A(2,2,J) = CP + SP\*TB

A(2,3,J) = SP + (1.-CP)\*TB

```

A(3,3,J) = 1.
A(4,1,J) = SP + (1.-CP)*TA
A(4,2,J) = RO*(1.-CP)
A(4,3,J) = RO*(PR-SP)
A(4,4,J) = 1.
A(5,5,J) = 1.-PR*TA
A(5,6,J) = PR*RO
A(6,5,J) = (-TA-TB+PR*TA*TB)/RO
A(6,6,J) = 1.-PR*TB
DIST(J) = A(5,6,J)
RETURN
END

```

# SUBROUTINE QUAD(J,EL,FG)

```

C
C . . . . COMPUTE QUAD MATRIX FROM INPUT PARAMETERS
COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL
C . . . . . SET ELEMENTS USED IN MAG OR ROT TO ZERO
A(1,3,J) = 0.
A(1,5,J) = 0.
A(2,3,J) = 0.
A(2,6,J) = 0.
A(4,1,J) = 0.
A(4,2,J) = 0.
A(4,3,J) = 0.
A(5,1,J) = 0.
A(6,2,J) = 0.
C . . . . . COMPUTE MATRIX ELEMENTS
IF(FG) 100,200,300
100  FG = -FG
    EPL = EXP(FG*EL)
    EMI = EXP(-FG*EL)
    SF = SIN(FG*EL)
    A(1,1,J) = .5*(EPL + EMI)
    A(1,2,J) = (EPL-EMI)/(2.*FG)
    A(2,1,J) = FG*FG*A(1,2,J)
    A(5,5,J) = COS(FG*EL)
    A(5,6,J) = SF/FG
    A(6,5,J) = -FG*SF
    FG = -FG
    GO TO 400
300  SF = SIN(FG*EL)
    EPL = EXP(FG*EL)
    EMI = EXP(-FG*EL)

```

```

      A(1,1,J) = COS(FG*EL)
      A(1,2,J) = SF/FG
      A(2,1,J) = -FG*SF
      A(5,5,J) = .5*(EPL+EMI)
      A(5,6,J) = (EPL-EMI)/(2.*FG)
      A(6,5,J) = FG*FG*A(5,6,J)
      GO TO 400
200   A(1,1,J) = 1.
      A(1,2,J) = EL
      A(2,1,J) = 0.
      A(5,5,J) = 1.
      A(5,6,J) = EL
      A(6,5,J) = 0.
400   A(2,2,J) = A(1,1,J)
      A(3,3,J) = 1.
      A(4,4,J) = 1.
      A(6,6,J) = A(5,5,J)
      DIST(J) = EL
      RETURN
      END

```

#### SUBROUTINE ROT(J,AA)

```

C
C . . . .COMPUTE ROTATION MATRIX FROM INEUT PARAMATERS
COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL
C . . . . .SET ELEMENTS USED IN MAG OR QUAD TC ZERO
      A(1,2,J) = 0.
      A(1,3,J) = 0.
      A(2,1,J) = 0.
      A(2,3,J) = 0.
      A(4,1,J) = 0.
      A(4,2,J) = 0.
      A(4,3,J) = 0.
      A(5,6,J) = 0.
      A(6,5,J) = 0.
C . . . . .COMPUTE MATRIXELEMENTS
      AR = AA*3.141593/180.
      CS = CCS(AR)
      SN = SIN(AR)
      A(1,1,J) = CS
      A(1,5,J) = SN
      A(2,2,J) = CS
      A(2,6,J) = SN
      A(5,1,J) = - SN

```

```

A(5,5,J) = CS
A(6,2,J) = -SN
A(6,6,J) = CS
A(4,4,J) = 1.
A(3,3,J) = 1.
DIST(J) = 0.
RETURN
END

```

SUBROUTINE DRIFT(DEL,ZTCT,IZ)

```

C
C . . . .TRANSPORT BEAM THRU DRIFT TUBE, COMPUTE POSITIONS
DIMENSION SIG1(6,6),SIG2(6,6),XMEX(6)
COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL
COMMON/DSPXY/XDSP(150),YDSP(150,2),JMAX
COMMON/TAX/T(6,6),XMAX(6)
COMMON/SAV/TSV(6,6,15),ZNCT(16)
EQUIVALENCE (A(1,1,15),SIG1(1,1))
EQUIVALENCE (TSV(1,1,15),SIG2(1,1))
ZINT=ZTOT
DO 215 M1=1,6
DO 215 M2=1,6
SIG1(M1,M2)=0.
DO 205 M3=1,6
SIG1(M1,M2)=SIG1(M1,M2)+T(M1,M3)*(XMAX(M3)**2)*T(M2,M3)
205 CONTINUE
SIG2(M1,M2)=SIG1(M1,M2)
215 CONTINUE
DO 255 K1=1,10
DO 225 M1=1,6
SIG2(1,M1)=SIG2(1,M1)+DEL*SIG1(2,M1)
SIG2(5,M1)=SIG2(5,M1)+DEL*SIG1(6,M1)
225 CONTINUE
DO 235 M1=1,6
SIG2(M1,1)=SIG2(M1,1)+DEL*SIG1(M1,2)
SIG2(M1,5)=SIG2(M1,5)+DEL*SIG1(M1,6)
235 CONTINUE
SIG2(1,1)=SIG2(1,1)+(DEL**2)*SIG1(2,2)
SIG2(1,5)=SIG2(1,5)+(DEL**2)*SIG1(2,6)
SIG2(5,1)=SIG2(5,1)+(DEL**2)*SIG1(6,2)
SIG2(5,5)=SIG2(5,5)+(DEL**2)*SIG1(6,6)
IF(IZ.EQ.999) GO TO 260
ZINT=ZINT+DEL
IZ=IZ+1

```

```

      YDSP (IZ, 1) = SQRT (SIG2 (1, 1))
      XDSP (IZ) = ZINT
      YDSP (IZ, 2) = SQRT (SIG2 (5, 5))
      DO 245 M1=1,6
      DO 245 M2=1,6
      SIG1 (M1, M2) = SIG2 (M1, M2)
245    CONTINUE
255    CONTINUE
      RETURN
260    DO 265 ID=1,6
265    XMEX (ID) = SQRT (SIG2 (ID, ID))
      WRITE (6, 10) (XMEX (IW), IW=1, 6)
10    FORMAT (1H0, 6HXMAX= , 6 (3X, F7.3))
      RETURN
      END

```

#### SUBROUTINE MATMLT (JJ)

```

C
C . . . . ROUTINE FOR MATRIX MULTIPLICATION
      DIMENSION C (6, 6)
      COMMON /ADZI/ A (6, 6, 15), DIST (15), ZL (16), IEL
      COMMON /TAX/ T (6, 6), XMAX (6)
      EQUIVALENCE (A (1, 1, 15), C (1, 1))
      DO 105 M1=1,6
      DO 105 M2=1,6
      C (M1, M2) = 0.
      DO 105 M3=1,6
      C (M1, M2) = A (M1, M3, JJ) * T (M3, M2) + C (M1, M2)
105    CONTINUE
      DO 115 M1=1,6
      DO 115 M2=1,6
      T (M1, M2) = C (M1, M2)
115    CONTINUE
      RETURN
      END

```

#### SUBROUTINE DSPLA

```

C
C . . . . . THIS ROUTINE USES SCOPE DISPLAY
      INTEGER XYP (2, 320), YPX, YPY, V (5, 2), P (2, 2), TAGS (4, 30)
      LOGICAL BUTTNS, PENSW
      DIMENSION RJ (15), RL (4)

```

```

COMMON/TAG/RTAGS(2,30),XYP
COMMON/DSPLY/XDSP(150),YDSP(150,2),JMAX
COMMON/ELEMT/RO(15),PHI(15),ALF(15),BET(15),EL(15),FG(15),
1ALPH(15),IQMR(15)
EQUIVALENCE (TAGS(1,1),RTAGS(1,1))
DATA RL(1),RL(2),RL(3),RL(4)/1HM,1HQ,1HR,1HT/
DATA RJ(1),RJ(2),RJ(3),RJ(4),RJ(5),RJ(6),RJ(7),RJ(8),RJ(9),
1RJ(10),RJ(11),RJ(12),RJ(13),RJ(14),RJ(15)/1H1,1H2,1H3,1H4,
21H5,1H6,1H7,1H8,1H9,2H10,2H11,2H12,2H13,2H14,2H15/
C.....Z SCALE, 1 POINT/INCH
C.....VERTICAL SCALE, 1 POINT=0.005 INCH (1.5"=300)
100      NP=JMAX*10
        DO 125 K=1,NP
        KY=2*K
        KX=KY-1
        XYP(2,KX)=512.+200.*YDSP(K,1)
        XYP(2,KY)=512.-200.*YDSP(K,2)
        XPZ=XDSP(K)
        XYP(1,KY)=XPZ
125      XYP(1,KX)=XPZ
        NP2=2*NP
        NX=NP2 + 1
        NY=NX + 1
        XYP(2,NX)= 512. + 1000.*YDSP(NP,1)
        XYP(2,NY)= 512. - 1000.*YDSP(NP,2)
        YPX = XYP(2,NX)
        YPY = XYP(2,NY)
        XYP(1,NX) = 1000
        XYP(1,NY) = 1022
C . . . . .GENERATE 1 MM MARKERS
200      NM1 = NY+1
        NM2 = NM1 + 9
        MM = 40
        NZ = 1010
        DO 225 L=NM1,NM2
        XYP(2,L) = 512 + MM
        XYP(1,L) = NZ
        MM = - MM
225      NZ = NZ + 1
        NP2 = NP2 + 12
        CALL VPPNTS(XYP,NP2,IHIT,1)
C . . . . .GENERATE BOUNDARIES AT +1.5",-1.5"
300      MINUS = -1
        DO 325 LL=1,2
        P(1,1) = 0
        P(1,2) = 950

```

```

MP = 512 + MINUS*300
P(2,1) = MP
P(2,2) = MP
CALL VECTOR(P,V,LL)
325 MINUS = - MINUS
390 CALL VPVECT(V,2,IH,2)
400 MQ = 1
DO 455 J=1,JMAX
JE = 2*J
JO = JE-1
RTAGS(1,JO) = RJ(J)
K = (IQMR(J) + 1) / 2
RTAGS(1,JE) = RL(K)
410 J10 = 10*J
TAGS(3,JO) = XDSP(J10)
TAGS(3,JE) = XDSP(J10)
IF(IQMR(J).NE.3) GO TO 420
MQLO = 0
IF(MQ.EQ.-1) MQLO = 30
TAGS(4,JO) = 512 + MQ*(305 + MQLO)
TAGS(4,JE) = 512 + MQ*(305 + 30 + MQLO)
MQ = - MQ
GO TO 455
420 TAGS(4,JO) = 512 + 305
TAGS(4,JE) = 512 + 305 + 30
455 CONTINUE
NT = JMAX * 2
490 CALL VPTAGS(TAGS,NT,IT,3)
500 CALL VPSTPT
CALL CLKON
IF(YPX.GT.552.OR.YPY.IT.472) GO TO 526
DO 525 LA=1,7
525 CALL LAMPON(LA)
526 CALL LAMPCN(2)
IBUT=0
530 IF(PENSW(.TRUE.)) GO TO 600
IF(BUTINS(IBUT)) GO TO 530
IF(IBUT.LT.128) GO TO 530
DO 535 LA=1,7
535 CALL LAMPCF(LA)
RETURN
600 NH=(IHIT+1)/2
CALL LANDT(NH)
GO TO 530
END

```

# SUBROUTINE OUTPT

```

C
C . . . . WRITE OUT DISPLAY PCINTS
COMMON/DSPXY/XDSP(150),YDSP(150,2),JMAX
WRITE(6,10)
10  FORMAT(1H0,5X,45HX AND Y BEAM CCGCRDS AS FUNCTION OF DISTANCE Z/)
    J3 = IROTOR(3)
    J4 = IROTOR(4)
    JBGH = 10*J3 - 9
    JEND = 10*J4
    IF(JEND.GT.140) JEND=140
    DO 115 J1=JBGH,JEND,10
      J2=J1+9
      WRITE(6,20) (YDSP(JX,1),JX=J1,J2)
20  FORMAT(2H X,10F7.3)
      WRITE(6,30) (XDSP(JZ),JZ=J1,J2)
30  FORMAT(1H ,10F7.2)
      WRITE(6,40) (YDSP(JY,2),JY=J1,J2)
40  FORMAT(2H Y,10F7.3///)
115 CONTINUE
    RETURN
    END

```

# SUBROUTINE REPET(JMIN)

```

C
C . . . . CHANGES SPECIFIED ELEMENTS AND REPEATS TRANSPORT
          LOGICAL AGAIN,TOGGLS,BITEST
COMMON/ELEMT/RO(15),PHI(15),ALP(15),BET(15),EL(15),FG(15),
          1ALPH(15),IQMR(15)
COMMON/DSPXY/XDSP(150),YDSP(150,2),JMAX
C
C . . . . .IF TOGGL 11 IS SET, GET NEW QUAD CURRENT FROM THUMB
CALL TOGGLS(ITOG)
IF(.NOT.BITEST(ITOG,11)) GO TO 301
J1=IROTOR(1)
J2=IROTOR(2)
JMIN = J1
IF(J2.LT.J1) JMIN=J2
IFG1=ITHUMB(1)
IFG2=ITHUMB(2)
FG1=FLOAT(IFG1)*0.00001
FG2=FLOAT(IFG2)*0.00001
IF(BITEST(ITOG,13)) FG1=-FG1
IF(BITEST(ITOG,17)) FG2=-FG2

```

```

      FG(J1) = FG1
      FG(J2) = FG2
      EL1=EL(J1)
      EL2=EL(J2)
      WRITE(6,10) J1,FG1,J2,FG2
10    FORMAT(1H0,6HJ, FG,5X,2(I2,F10.5))
      CALL QUAD(J1,EL1,FG1)
      CALL QUAD(J2,EL2,FG2)
      IF(.NOT.BITEST(ITOG,10)) GO TO 500
      CALL RITE(J1)
      CALL RITE(J2)
      GO TO 500

C
C . . . . .INPUT ELEMENT PARAMETERS THRU TELETYPE
301   CALL INPUT1(JMAX,4HJMAX,1)
      JMIN = JMAX
302   CALL INPUT1(JJ,5HNELEM,1)
      IF(JJ.LT.JMIN) JMIN=JJ
      CALL REED(JJ)
      CALL TOGGLS(ITOG)
      IF(.NOT.BITEST(ITOG,10)) GO TO 303
      CALL RITE(JJ)
303   CALL INPUT1(AGAIN,5HAGAIN,4)
      IF(AGAIN) GO TO 302
500   RETURN
      END

```

#### SUBROUTINE RFRESH

```

C
C . . . . .WRITE PARAMETERS AFTER RESTAET FROM PREEVIOUS DUMP
      LOGICAL PRINP
      COMMON/ELEMT/RO(15),PHI(15),ALP(15),BET(15),EL(15),FG(15),
      1ALPH(15),IQMR(15)
      COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL
      COMMON/DSPXY/XDSP(150),YDSP(150,2),JMAX

C
      WRITE(6,20)
20    FORMAT(42H0 DO YOU WISH TO PRINT CURRENT PARAMETERS?)
      CALL INPUT1(PRINP,5HPRINT,4)
      IF(.NOT.PRINP) GO TO 500
      DO 295 L=1,JMAX
      LW=L
      WRITE(6,40) LW,IQMR(L),ZL(L),DIST(L)
40    FORMAT(1H0,17H ELEMENT NUMBER =,I2,6H TYPE ,I1,12H
      PRE-DRIFT= ,

```

```

1F10.2,11H   LENGTH=,F10.2)
IW=IQMR(L)
GO TO (231,231,233,233,235,235,500),IW
231 WRITE(6,10) RO(L),PHI(L),ALF(L),BET(L)
10  FORMAT(1H0,6HRHO =,F10.2,6HPHI =,F10.2,8HALPHA=,F10.2,7H BE
    1TA =,F10.2/)
GO TO 295
233 WRITE(6,30) FG(L)
30  FORMAT(1H0,11H FLD GRAD =,F10.5/)
GO TO 295
235 WRITE(6,50) ALPH(L)
50  FORMAT(1H0,17H ROTATION ANGLE =,F10.2/)
295 CONTINUE
500 RETURN
END

```

#### SUBROUTINE LANDT(NH)

```

C
C . . . . . CALCULATE L FROM XMAX; CALCULATE T MATRIX
COMMON/ADZI/A(6,6,15),DIST(15),ZL(16),IEL
COMMON/TAX/T(6,6),XMAX(6)
COMMON/ELEMT/RO(15),PHI(15),ALF(15),BET(15),EL(15),FG(15),
    1ALPH(15),IQMR(15)
COMMON/SAV/TSV(6,6,15),ZNCT(16)
C . . . . . JH IS INDEX FOR HIT
100 JH=(NH-1)/10+ 1
    JHM1 = JH - 1
    DEL=ZL(JH)/10.
    IZL= NH-JHM1*10
    ZLH = DEL*FLOAT(IZL)
    IF(JH.NE.1) GO TO 250
C . . . . . SET UP SPECIAL CASE FOR JH=1
    JHM1=15
210 DO 215 M1=1,6
    DO 215 M2=1,6
    TSV(M1,M2,15) = 0.
215 IF(M1.EQ.M2) TSV(M1,M2,15) = 1.
C . . . . . STORE TRANSECT MATRIX IN A(I,M,15)
250 DO 255 L=1,6
    A(1,L,15) = TSV(1,L,JHM1) + TSV(2,L,JHM1)*ZLH
    A(2,L,15) = TSV(2,L,JHM1)
    A(3,L,15) = TSV(3,L,JHM1)
    A(4,L,15) = TSV(4,L,JHM1)
    A(5,L,15) = TSV(5,L,JHM1) + TSV(6,L,JHM1)*ZLH

```

```

255      A(6,L,15) = TSV(6,L,JHM1)
C . . . . .SET UP PARAMETERS FOR CALL TO RITE
      DIST(15) = ZLH
      IQMR(15) = 9
      ZL(15) = ZLH
      IF(JH.NE.1) ZL(15) = ZL(15)+ZNOT(JH)
      CALL RITE(15)
C . . . . .SET UP T MATRIX FOR CALL TO DRIFT
      DO 305 M1=1,6
      DO 305 M2=1,6
305      T(M1,M2) = TSV(M1,M2,JHM1)
      DEL=ZLH
      CALL DRIFT(DEL,0.,999)
      RETURN
      END

```

## APPENDIX B

### IBM 7094 VERSION OF BEAMR

The IBM 7094 version of the BEAMR program is similar to that just described except that there exists no interactive mode and no facility for display of results. Rather, the program supplies printed description of the beam and cumulative values for the transfer matrix at points specified. In addition, the program will automatically search to determine what values of the quadrupole focusing strength are required to obtain preset values for two elements of the total transfer matrix at a preset position.

The input for the program is as follows:

CARD NUMBER	FORMAT	INFORMATION
1	13A6	Title
2	6F10.0	$x_{\max}, x'_{\max}, \Delta p/p, \Delta\lambda, y_{\max}, y'_{\max}$
3	2I2	JMIX, JMOX
4	1F10.0	DELK
5	5F10.0, I1	P1, P2, P3, P4, P5, XJ
-	-	-
-	-	-
-	-	-
4+JMIX	-	-
4+JMIX+1	4I3, 4E15.8, 2I3	AI, AJ, BI, BJ, SFOC1, SFOC2, VAL1, VAL2, J1, J2
	(This card only for JMOX=1.)	

#### Definitions of Input

The parameters  $x_{\max}, x'_{\max}, \Delta p/p, \Delta\lambda, y_{\max}, y'_{\max}$  are characteristics of the beam at the entrance to the system, as defined previously, in inches and radians. JMIX is the total number of beam-handling elements in the system, including drift spaces. JMOX specifies search procedures: 0, no search; 1, search on quadrupole strengths.

The total transfer matrix and the maximum beam dimensions in the drift space beyond the last beam-handling element will be calculated and printed out 25 times

if so desired. These calculations will be done at intervals DELK (in.) from the exit of this element.

The definitions of P1, P2, P3, P4, and P5 vary depending on the type of beam-handling element involved. The type of element is defined by XJ:

For a magnet, XJ=1:

P1      radius of curvature  
P2      angle of deflection  
P3      entrance edge angle  
P4      exit edge angle  
P5      PL

For a drift space, XJ=2:

P1      length of drift space  
P2      0  
P3      0  
P4      0  
P5      0

For a quadrupole magnet, XJ=3,:

P1      effective length of quadrupole  
P2      strength, k - positive for horizontal focusing, negative for vertical focusing  
P3      PL  
P4      0  
P5      0

For a quadrupole (part of a symmetric triplet), XJ=4: P1,P2,P3,P4, and P5 are the same as for XJ=3. For a rotation, XJ=5: P1 is the magnitude and direction of the rotation - positive for a clockwise rotation, negative for a counterclockwise rotation.

The quantity PL defines, for each beam-handling element, whether a description of the beam for the drift space following the element will be printed out. If PL = 1, the drift space following the element will be divided into 20 equal intervals and a description of the beam (total  $\mathcal{T}$  matrix and maximum beam dimensions) will be printed as calculated at the end of each interval. If PL = 0, no output is printed.

The quantities AI,AJ,BI,BJ,SFOC1,SFOC2,VAL1,VAL2,J1, and J2 specify search procedures. The search routine will attempt to adjust the strengths of quadrupoles J1 and J2 so as to produce, at a distance SFOC1 beyond the last element, the value VAL1 for the transfer matrix element T(AI,AJ). Simultaneously, it will attempt

to produce at the distance SFOC2 beyond the last element the value VAL2 for the transfer matrix element T(BI,BJ) .

If one of the quadrupoles specified is the first section of a symmetric triplet (XJ=4) , the focusing strength of the last section of the triplet will be held identically equal to that of the first section during all search procedures .

The search routine optimizes the specified parameters by first constructing numerically the gradients

$$\begin{array}{cc} \frac{\partial T(AI, AJ)}{\partial k(J1)} & \frac{\partial T(AI, AJ)}{\partial k(J2)} \\ \frac{\partial T(BI, BJ)}{\partial k(J1)} & \frac{\partial T(BI, BJ)}{\partial k(J2)} \end{array}$$

and then solving the equations

$$\begin{aligned} \Delta T(AI, AJ) &= \left[ \frac{\partial T(AI, AJ)}{\partial k(J1)} \right] \Delta k(J1) + \left[ \frac{\partial T(AI, AJ)}{\partial k(J2)} \right] \Delta k(J2) \\ \Delta T(BI, BJ) &= \left[ \frac{\partial T(BI, BJ)}{\partial k(J1)} \right] \Delta k(J1) + \left[ \frac{\partial T(BI, BJ)}{\partial k(J2)} \right] \Delta k(J2) \end{aligned}$$

for corrections  $\Delta k$  to the field gradients . The quantity  $\Delta T(AI, AJ)$  is the difference between the current value of the (AI,AJ) matrix element and the desired value VAL1 .

## APPENDIX C

### FORTRAN LISTING OF IBM 7094 VERSION

\$IPFTC BEAMR DECK

```

C      BEAMR DOES BEAM TRANSPORT CALCULATIONS IN FIRST ORDER, USING
C      A BEAM TRANSFER MATRIX. IT TREATS MAGNETS, DRIFT SPACES, QUADRUPOLE
C      LENSES, AND ROTATIONS.
C      INPUT IS AS FOLLOWS
C      CARD ONE CONTAINS A TITLE
C      CARD 2 (6F10.0) CONTAINS A DESCRIPTION OF THE BEAM AT THE SYSTEM
C      ENTRANCE
C          A) MAXIMUM SIZE OF BEAM IN THE X-DIRECTION (INCHES)
C          B) MAXIMUM DIVERGENCE OF THE BEAM IN THE X PLANE (RADIAN)
C          C) INCIDENT MOMENTUM SPREAD (FRACTIONAL)
C          D) MAXIMUM SPREAD IN PATH LENGTH ENTERING THE SYSTEM (INCHES)
C          E) MAXIMUM EXTENSION OF THE BEAM IN THE Y DIRECTION (INCHES)
C          F) MAXIMUM DIVERGENCE OF THE BEAM IN THE Y PLANE (RADIAN)
C      CARD 3 (FORMAT 2I2) CONTAINS JMIK AND JMOX
C          JMIK IS THE TOTAL NUMBER OF ELEMENTS (DRIFT SPACES, QUADRUPOLE
C          SINGLETS, MAGNETS, ROTATIONS) IN THE BEAM LINE
C          JMOX SPECIFIES SEARCH PROCEDURES
C              JMOX=0 IMPLIES NO SEARCH
C              JMOX=1 IMPLIES SEARCHING ON QUADRUPOLE CURRENTS
C      CARD 4 (1F10.0) CONTAINS THE QUANTITY DELK (INCHES)
C      CARDS 5 TO 5+JMIK (5F10.0, 1I1) DESCRIBE INDIVIDUAL BEAM ELEMENTS,
C      ONE CARD PER ELEMENT
C          MAGNET- COL 1-10 CONTAIN THE RADIUS OF CURVATURE (GREATER THAN
C                      ZERO)
C          COL 11-20 CONTAIN THE ANGLE OF DEFLECTION (GREATER THAN
C                      ZERO)
C          COL 21-30 CONTAIN THE ENTRANCE ANGLE (POSITIVE ANGLES
C                      YIELD VERTICAL FOCUSING)
C          COL 31-40 CONTAIN THE EXIT ANGLE ( POSITIVE ANGLES
C                      YIELD VERTICAL FOCUSING)
C          COL 41-50 CONTAIN PL
C          COL 51 CONTAINS XJ (SHOULD EQUAL 1 FOR A MAGNET)
C          THE ABOVE SIGN CONVENTIONS ARE FOR A MAGNET WHICH
C          BENDS TO THE RIGHT. FOR A LEFT-BENDING MAGNET ALL
C          SIGNS ARE REVERSED.
C      QUADRUPOLE COL 1-10 EFFECTIVE LENGTH OF THE QUAD (INCHES)
C          COL 11-20 CONTAIN K, WHERE K IS THE QUADRUPOLE

```

```

C          STRENGTH DEFINED BY
C          K=SQRT(FIELD GRADIENT(KG/INCH)/BR(KG-INCHES))
C          POSITIVE K YIELDS X-FOCUS, Y-DEFOCUS
C          NEGATIVE K YIELDS X-DEFOCUS, Y-FOCUS
C          COL 21-30 CONTAIN FL
C          COL 51 CONTAINS XJ (SHOULD BE 3 FOR A QUADRUPOLE)
C          IF COL 51 CONTAINS A 4 IT INDICATES THAT THE ELEMENT
C          BEING DESCRIBED IS TO BE CONSIDERED AS THE FIRST
C          SECTION OF A SYMMETRIC TRIPLET, AND IN ANY SEARCH
C          PROCEDURE THE LAST ELEMENT OF THE TRIPLET WILL ALWAYS
C          HAVE A STRENGTH EQUAL TO THIS ONE
C          DRIFT SPACE COL 1-10 LENGTH OF THE DRIFT (INCHES)
C          COL 11-21 PL (SHOULD BE ZERO)
C          COL 51 XJ (SHOULD BE 2 FOR A DRIFT SPACE)
C          ROTATION, COL 1-10 CONTAIN THE MAGNITUDE OF THE ROTATION
C          IN DEGREES. . A POSITIVE ROTATION INDICATES A
C          DOWNWARD BEND, A NEGATIVE ROTATION INDICATES
C          AN UPWARD BEND
C          COL 17-20 CONTAIN PL
C          COL 51 CONTAINS XJ (SHOULD BE 5 FOR A ROTATION)
C          FOR ANY ELEMENT, SETTING PL=1 CAUSES THE TRANSFER MATRIX TO BE
C          PRINTED OUT AT INTERVALS OF DELTA INCHES BEYOND THE GIVEN
C          ELEMENT, WHERE DELTA=1/20 OF THE FOLLOWING DRIFT SPACE. FOR THE
C          LAST ELEMENT THE TRANSFER MATRIX IS PRINTED OUT 25 TIMES IN
C          INTERVALS OF DELK.
C          IF PL=0, THE TRANSFER MATRIX IS NOT PRINTED AT THE EXIT OF
C          THE GIVEN ELEMENT.
C          LAST CARD (4I3,4F15.8,2I3)
C          CONTAINS AI,AJ,BI,BJ,SFOC1,SFOC2,VAL1,VAL2,J1,J2.
C          THE SEARCH ROUTINE WILL ATTEMPT TO VARY THE STRENGTHS OF THE
C          J1-TH AND J2-TH QUADRUPOLES IN ORDER TO MAKE THE (AI,AJ)
C          AND (BI,BJ) ELEMENTS OF THE TRANSFER MATRIX EQUAL TO THE VALUES
C          VAL1 AND VAL2 RESPECTIVELY AT DISTANCES SFOC1 AND SFOC2
C          RESPECTIVELY BEYOND THE LAST BEAM ELEMENT.

```

```

C          INTEGER X,F
C          DIMENSION X(50),RO(50),PHI(50),ALP(50),BET(50),ALPR(50),PHIR(50), -
C          1BETR(50),A(6,6,50),Z(50),T(6,6),TITLE(13),BINT1(3),RINT2(3), -
C          2FG(50),EL(50),GRAD(2,2),TINT1(6,6,3),TINT2(6,6,3),PL(50),ALPH(50), -
C          3 ALPHR(50),A1(6,6),A2(6,6),XMAX(6),AC(5,50)
C          COMMON A,T,XMAX

```

```

100 READ(5,1) TITLE
    READ(5,2) (XMAX(I),I=1,6)
    READ(5,3) JMIX,JMOX
    READ(5,2) DELK

```

**Page Intentionally Left Blank**

```

ALPHR(J)=ALPH(J)*3.14159265/180.
A(1,1,J)=COS(ALPHR(J))
A(1,5,J)=SIN(ALPHR(J))
A(2,2,J)=COS(ALPHR(J))
A(2,6,J)=SIN(ALPHR(J))
A(5,1,J)=-SIN(ALPHR(J))
A(5,5,J)=COS(ALPHR(J))
A(6,2,J)=-SIN(ALPHR(J))
A(6,6,J)=COS(ALPHR(J))
A(4,4,J)=1.0
A(3,3,J)=1.0
WRITE(6,15) J,ALPH(J)

199 WRITE(6,10) (A(1,N,J),N=1,6)
   WRITE(6,10) (A(2,N,J),N=1,6)
   WRITE(6,10) (A(3,N,J),N=1,6)
   WRITE(6,10) (A(4,N,J),N=1,6)
   WRITE(6,10) (A(5,N,J),N=1,6)
   WRITE(6,10) (A(6,N,J),N=1,6)
200 CONTINUE

203 DO 201 M1=1,6
   DO 201 M2=1,6
   T(M1,M2)=A(M1,M2,1)
201 CONTINUE

   DO 300 J=2,JMIX
   CALL MATMLT(J)
   IF(X(JMAX).NE.0) GO TO 300
   IF(PL(J).EQ.0.0) GO TO 300
   WRITE(6,14) J,X(J)
   DEL=(Z(J+1)/20.0)+1.0
   IDEL=DEL
   DEL=IDEL
   IF(J+1.EQ.JMAX) DEL=DELK
   CALL EXIT(DEL)
300 CONTINUE

401 IF(X(JMAX)) 100,100,998
998 IF(F-2) 999,1000,1000
999 IF(ZX) 9995,9995,1000
9995 READ(5,11) IA,JA,IB,JB,SFOC1,SFCC2,VAL1,VAL2,J1,J2
   WRITE(6,11) IA,JA,IB,JB,SFOC1,SFOC2,VAL1,VAL2,J1,J2

1000 A1(1,1)=1.0

```

```

A1(1,2)=SFOC1
A1(2,2)=1.0
A1(3,3)=1.0
A1(4,4)=1.0
A1(5,5)=1.0
A1(5,6)=SFOC1
A1(6,6)=1.0

```

```

DO 1001 L1=1,6
DO 1001 L2=1,6
TINT1(L1,L2,F)=0.0
DO 1001 L3=1,6
TINT1(L1,L2,F)=TINT1(L1,L2,F)+A1(L1,L3)*T(L3,L2)

```

1001 CONTINUE

```

WRITE(6,75) F, (TINT1(1,N1,F), N1=1,6)
WRITE(6,75) F, (TINT1(2,N1,F), N1=1,6)
WRITE(6,75) F, (TINT1(3,N1,F), N1=1,6)
WRITE(6,75) F, (TINT1(4,N1,F), N1=1,6)
WRITE(6,75) F, (TINT1(5,N1,F), N1=1,6)
WRITE(6,75) F, (TINT1(6,N1,F), N1=1,6)
A2(1,1)=1.0
A2(1,2)=SFOC2
A2(2,2)=1.0
A2(3,3)=1.0
A2(4,4)=1.0
A2(5,5)=1.0
A2(5,6)=SFOC2
A2(6,6)=1.0

```

```

DO 21002 K1=1,6
DO 21002 K2=1,6
TINT2(K1,K2,F)=0.0
DO 21002 K3=1,6
TINT2(K1,K2,F)=TINT2(K1,K2,F)+A2(K1,K3)*T(K3,K2)

```

21002 CONTINUE

```

WRITE(6,75) F, (TINT2(1,N1,F), N1=1,6)
WRITE(6,75) F, (TINT2(2,N1,F), N1=1,6)
WRITE(6,75) F, (TINT2(3,N1,F), N1=1,6)
WRITE(6,75) F, (TINT2(4,N1,F), N1=1,6)
WRITE(6,75) F, (TINT2(5,N1,F), N1=1,6)
WRITE(6,75) F, (TINT2(6,N1,F), N1=1,6)

```

```

2000 DIFF1=VAL1-TINT1(IA,JA,1)
DIFF2=VAL2-TINT2(IB,JB,1)

```

**Page Intentionally Left Blank**

```

7  FORMAT(/,3H J=,I2,3X,4HRRHO=,1F10.4,3X,4HPhi=,F10.4,3X,6HALPHA=,
1F10.4,3X,5HEETA=,F10.4)
8  FORMAT(/,3H J=,I2,3X,7H DRIFT=,F10.4)
9  FORMAT(/,3H J=,I2,3X,17HEFFECTIVE LENGTH=,F10.4,5X,2HK=,F10.4)
10 FORMAT(10X,6F15.6)
11 FORMAT(4I3,4E15.6,2I3)
12 FORMAT (10X,8H G(1,1)=,1PE15.6,5X,8H G(1,2)=,1PE15.6)
13 FORMAT (10X,8H G(2,1)=,1PE15.6,5X,8H G(2,2)=,1PE15.6)
14 FORMAT(1H1,1X,2HJ=,I2,5X,5HX(J)=,I1)
15 FORMAT(/,3H J=,I2,3X,11HROTATION OF,1F8.3,7HDEGREES)
16 FORMAT(5F10.5,I1)
75 FORMAT(1X,2HF=,1I2,6F15.6)
END

```

```

$IEFTC QUAD6    LECK
SUBROUTINE QUAD(J,EL,FG)
DIMENSION A(6,6,50),T(6,6),XMAX(6)
COMMON A,T,XMAX
IF(FG) 100,200,300
100 FG=-FG
  A(1,1,J)=0.50*(EXP(FG*EL)+EXP(-FG*EL))
  A(1,2,J)=(EXF(FG*EL)-EXF(-EL*FG))/(2.0*FG)
  A(1,3,J)=0.0
  A(2,1,J)=FG*FG*A(1,2,J)
  A(2,2,J)=A(1,1,J)
  A(2,3,J)=0.0
  A(3,3,J)=1.0
  A(4,4,J)=1.0
  A(5,5,J)=COS(FG*EL)
  A(5,6,J)=SIN(FG*EL)/FG
  A(6,5,J)=-FG*SIN(FG*EL)
  A(6,6,J)=A(5,5,J)
  FG=-FG
  GO TO 400
300 A(1,1,J)=COS(FG*EL)
  A(1,2,J)=SIN(EL*FG)/FG
  A(2,1,J)=-FG*SIN(FG*EL)
  A(2,2,J)=A(1,1,J)
  A(3,3,J)=1.0
  A(4,4,J)=1.0
  A(5,5,J)=0.50*(EXP(EL*FG)+EXP(-EL*FG))
  A(5,6,J)=(EXF(FG*EL)-EXF(-FG*EL))/(2.0*FG)

```

```

      A(6,5,J)=FG*FG*A(5,6,J)
      A(6,6,J)=A(5,5,J)
      GO TO 400
200  A(1,1,J)=1.0
      A(1,2,J)=EL
      A(1,3,J)=0.0
      A(2,2,J)=1.0
      A(3,3,J)=1.0
      A(4,4,J)=1.0
      A(5,5,J)=1.0
      A(5,6,J)=EL
      A(6,6,J)=1.0
      GO TO 400
400  RETURN
      END

```

```

$IBFTC EXITU   DECK
      SUBROUTINE EXIT(DEL)
      COMMON A,T,XMAX
      DIMENSION A(6,6,50),T(6,6),TLAST(6,6),AL(6,6),XMAX(6),SIGM(6,6), -
1  SIG1(6,6),XMF(6),TLT(6,6)
      WRITE(6,1) DEL
      DO 200 N=1,25
      Y=N-1
      XL=Y*DEL
      AL(1,1)=1.0
      AL(1,2)=XL
      AL(2,2)=1.0
      AL(3,3)=1.0
      AL(4,4)=1.0
      AL(5,5)=1.0
      AL(5,6)=XL
      AL(6,6)=1.0
      DO 101 I1=1,6
      DO 101 I2=1,6
      TLAST(I1,I2)=0.0
      DO 101 I3=1,6
      TLAST(I1,I2)=TLAST(I1,I2)+AL(I1,I3)*T(I3,I2)
101  CONTINUE
      WRITE(6,3)
      WRITE(6,4) XL,(TLAST(1,K),K=1,6)
      WRITE(6,5)    (TLAST(2,K),K=1,6)

```

**Page Intentionally Left Blank**

PRECEDING PAGE BLANK NOT FILMED

```
      C(M1,M2)=A(M1,M3,JJ)*T(M3,M2)+C(M1,M2)
100  CONTINUE
      DO 101 M1=1,6
      DO 101 M2=1,6
      T(M1,M2)=C(M1,M2)
101  CONTINUE
      RETURN
      END
```

```
$IEFTC MAG6      DECK
      SUBROUTINE MAG(J,RO,PHI,ALP,BET)
      COMMON A,T,XMAX
      DIMENSION A(6,6,50),T(6,6),XMAX(6)
      PHIR=PHI*3.14159265/180.
      ALPR=ALP*3.14159265/180.
      BETR=BET*3.14159265/180.
      A(1,1,J)=COS(PHIR-ALPR)/COS(ALPR)
      A(1,2,J)=RO*SIN(PHIR)
      A(1,3,J)=RO*(1.0-COS(PHIR))
      A(2,1,J)=-SIN(PHIR-ALPR-BETR)/(RO*COS(ALPR)*COS(BETR))
      A(2,2,J)=COS(PHIR-BETR)/COS(BETR)
      A(2,3,J)=SIN(PHIR)+(1.0-COS(PHIR))*TAN(BETR)
      A(3,3,J)=1.0
      A(4,1,J)=SIN(PHIR)+(1.0-COS(PHIR))*TAN(ALPR)
      A(4,2,J)=RO*(1.0-COS(PHIR))
      A(4,3,J)=RO*(PHIR-SIN(PHIR))
      A(4,4,J)=1.0
      A(5,5,J)=1.0-PHIR*TAN(ALPR)
      A(5,6,J)=PHIR*RO
      A(6,5,J)=(-TAN(ALPR)-TAN(BETR)+PHIR*TAN(ALPR)*TAN(BETR))/RO
      A(6,6,J)=1.0-PHIR*TAN(BETR)
      RETURN
      END
```

## REFERENCES

1. Enge, H. A.: Focusing of Charged Particles. Albert Septier, ed., Academic Press, 1967, p. 203.
2. Steffen, Klaus G.: High Energy Beam Optics. Interscience Pub., 1965, pp. 1-16.
3. Brown, Karl L.; and Howry, Sam K.: TRANSPORT/360: A Computer Program for Designing Charged Particle Beam Transport Systems. Rep. SLAC-91, Stanford Univ., July 1970.
4. Fessler, Theodore E.: Magnetic-Field Measurements for the Lewis Research Center Cyclotron. NASA TM X-2692, 1973.
5. Bercaw, Robert W.; Fessler, Theodore E.; and Arnold, Jeffrey M.: A Programmable Computer Interface for CAMAC. NASA TN D-7148, 1973.

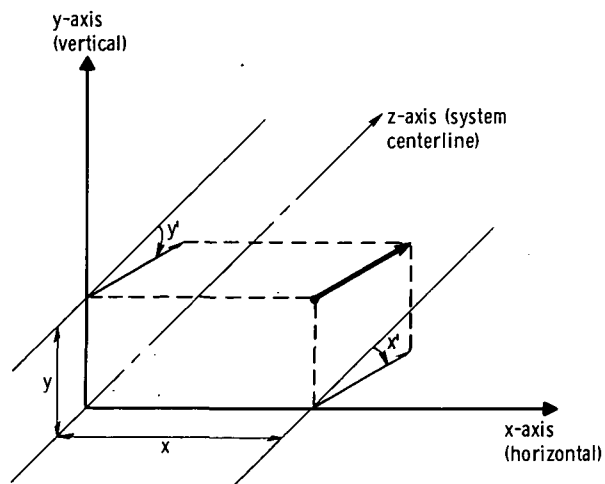


Figure 1. - Coordinate system used for BEAMR.

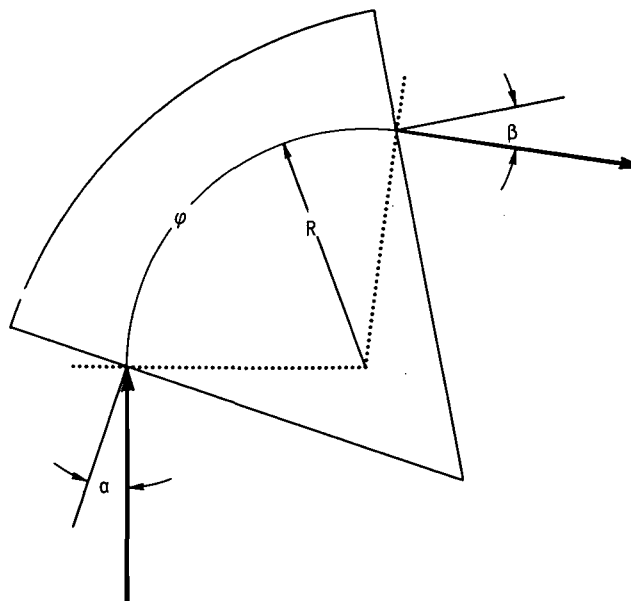


Figure 2 - Magnet dimensions. For right-bending magnet,  $R$  and  $\varphi$  are positive.  $\alpha$  and  $\beta$  (angles between beam and normal to pole face) are positive if (as shown here) center of curvature falls within magnet, negative if it falls outside.

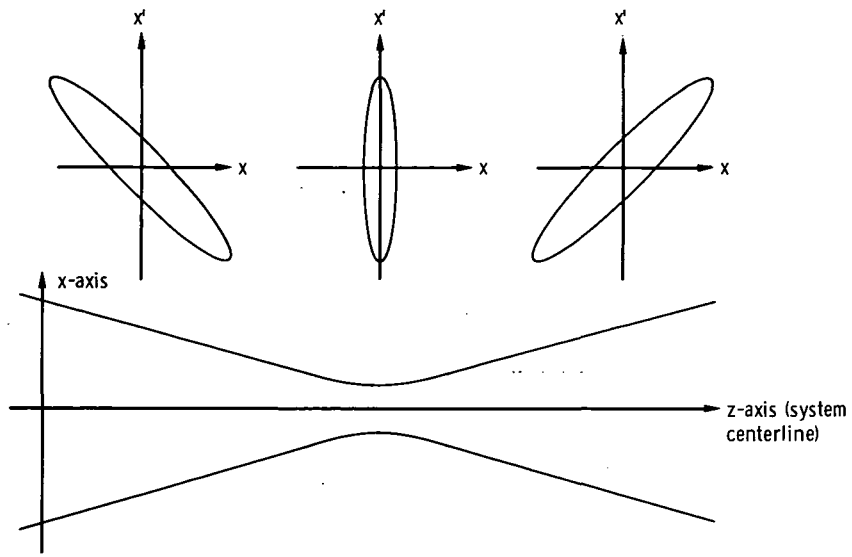


Figure 3. - Evolution of one-dimensional phase-space ellipse as beam envelope passes through a waist. Lower diagram indicates physical size of beam envelope over corresponding interval.

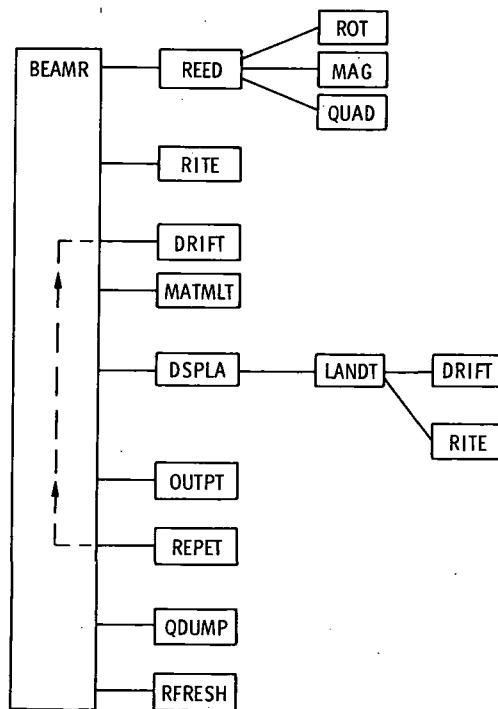


Figure 4. - Schematic of PDP-15 program flow.

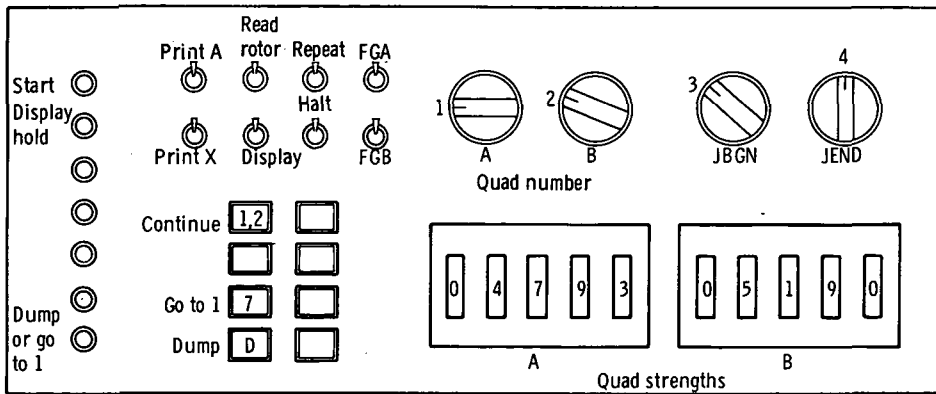


Figure 5. - Functional control panel.

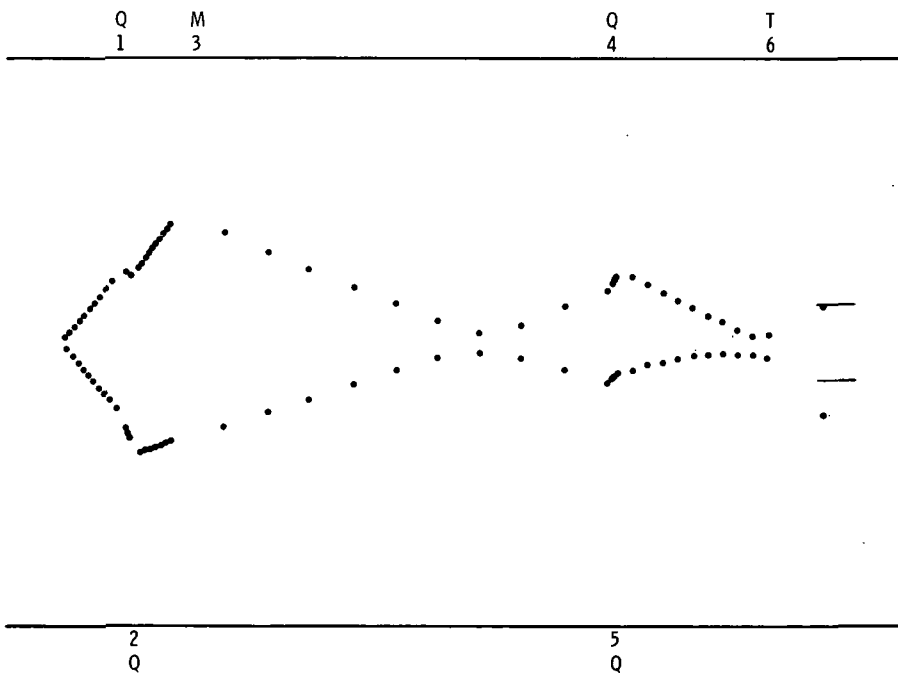


Figure 6. - Typical PDP-15 display of output from BEAMR. System features two sets of quadrupole doublets with a dipole magnet between them.



POSTMASTER: If Undeliverable (Section 158  
Postal Manual) Do Not Return

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

## NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

**TECHNICAL REPORTS:** Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

**TECHNICAL NOTES:** Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

**TECHNICAL MEMORANDUMS:** Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

**CONTRACTOR REPORTS:** Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

**TECHNICAL TRANSLATIONS:** Information published in a foreign language considered to merit NASA distribution in English.

**SPECIAL PUBLICATIONS:** Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

**TECHNOLOGY UTILIZATION PUBLICATIONS:** Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

*Details on the availability of these publications may be obtained from:*

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**  
**Washington, D.C. 20546**